Data Science & Python for Finance Stock Analysis

Final Project

Part 1 - Data Preparation and Exploration ¶

Creator: Wendy(Aobo) Liu

1.1 Download the constituents of the S&P 1500 using Capital IQ.

Packages and Settings

Configuration

In [2]:
```
! pip install intrinio_sdk
```

```
Requirement already satisfied: intrinio_sdk in /home/nbuser/anaconda3_501/lib/python3.6/site-packages (5.5.0)
Requirement already satisfied: six>=1.10 in /home/nbuser/anaconda3_501/lib/python3.6/site-packages (from intrinio_sdk)
(1.11.0)
Requirement already satisfied: certifi in /home/nbuser/anaconda3_501/lib/python3.6/site-packages (from intrinio_sdk) (20
18.10.15)
Requirement already satisfied: python-dateutil in /home/nbuser/anaconda3_501/lib/python3.6/site-packages (from intrinio_
sdk) (2.8.1)
Requirement already satisfied: urllib3>=1.15 in /home/nbuser/anaconda3_501/lib/python3.6/site-packages (from intrinio_sd
k) (1.23)
WARNING: You are using pip version 19.3.1; however, version 20.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

In [3]:
```
import intrinio_sdk
import configparser as cp
```

Scientific Analysis

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

Capital IQ - S&P Composite 1500 (^SP1500) > Constituents

In [5]:
```
sp_df = pd.read_excel("../data/SP1500 .xls", skiprows = 14, skipfooter = 12)
sp_df.head()
```

Out[5]:

	Company Name	Exchange:Ticker	Currency	Market Cap (mm) [Latest]*†	Revenue (mm)	% Price Change [Last Day]	% Price Change [30 Day]	% Price Change [YTD]	% Price Change [12 Month]	Price Close	P/E*†	P/BV*†	Primary Industry
0	3D Systems Corporation	NYSE:DDD	USD	975.90	629.09	5.66%	36.61%	(6.17%)	(26.37%)	8.21	NM	1.85x	Technology Hardware, Storage and Peripherals
1	3M Company	NYSE:MMM	USD	85359.14	32348	(0.13%)	10.92%	(15.88%)	(19.88%)	148.40	17.47x	8.41x	Industrial Conglomerates
2	8x8, Inc.	NYSE:EGHT	USD	1650.60	418.53	1.38%	14.58%	(11.97%)	(33.46%)	16.11	NM	7.40x	Application Software
3	A. O. Smith Corporation	NYSE:AOS	USD	6800.98	2992.7	0.79%	17.03%	(11.88%)	(21.74%)	41.98	18.75x	4.09x	Building Products
4	AAON, Inc.	NasdaqGS:AAON	USD	2363.66	469.33	-	1.11%	(8.10%)	(12.35%)	45.41	44.47x	8.15x	Building Products

In [6]:
```
sp_df.tail()
```

Out[6]:

	Company Name	Exchange:Ticker	Currency	Market Cap (mm) [Latest]*†	Revenue (mm)	% Price Change [Last Day]	% Price Change [30 Day]	% Price Change [YTD]	% Price Change [12 Month]	Price Close	P/E*†	P/BV*†	Primary Industry
1501	Zebra Technologies Corporation	NasdaqGS:ZBRA	USD	12145.76	4471	1.43%	28.46%	(10.44%)	10.81%	228.77	23.78x	7.03x	Electronic Equipment and Instruments
1502	Zimmer Biomet Holdings, Inc.	NYSE:ZBH	USD	23576.17	7982.2	(1.88%)	23.98%	(23.79%)	(8.24%)	114.07	21.21x	1.90x	Health Care Equipment
1503	Zions Bancorporation, National Association	NasdaqGS:ZION	USD	5025.34	2533	(0.74%)	24.27%	(40.93%)	(38.72%)	30.67	9.59x	0.73x	Regional Banks
1504	Zoetis Inc.	NYSE:ZTS	USD	60522.59	6260	(0.07%)	11.19%	(3.71%)	22.83%	127.44	40.89x	22.38x	Pharmaceuticals
1505	Zumiez Inc.	NasdaqGS:ZUMZ	USD	516.84	1034.13	(4.46%)	39.99%	(41.11%)	(24.50%)	20.34	8.00x	1.13x	Apparel Retail

In [7]:
```
sp_df.columns
```

Out[7]:
```
Index(['Company Name', 'Exchange:Ticker', 'Currency',
       'Market Cap (mm) [Latest]*†', 'Revenue (mm)',
       '% Price Change [Last Day]', '% Price Change [30 Day]',
       '% Price Change [YTD]', '% Price Change [12 Month]', 'Price Close',
       'P/E*†', 'P/BV*†', 'Primary Industry'],
      dtype='object')
```

1.2 Clean and organize the Excel file that you acquire from Capital IQ. Create a pandas data frame and save the clean file as sp1500.csv

In [8]:
```
sp_df.columns = ['company', 'Exchange:Ticker', 'currency',
                 'marketcap_mm', 'revenue_mm', 'pct_price_change_lastday',
                 'pct_price_change_30day', 'pct_price_change_ytd', 'pct_price_change_12_month',
                 'price_close', 'P/E*†', 'P/BV*†', 'industry', ]
```

In [9]:
```
sp_df['exchange'] = sp_df['Exchange:Ticker'].apply(lambda i:i.split(":")[0])
sp_df['ticker']  = sp_df['Exchange:Ticker'].apply(lambda i:i.split(":")[1])
sp_df = sp_df.drop(columns=['Exchange:Ticker','currency'])
```

Reorder columns

In [10]:
```
sp_df = sp_df[['company','ticker','price_close', 'pct_price_change_lastday',
               'pct_price_change_30day', 'pct_price_change_ytd', 'pct_price_change_12_month',
               'P/E*†', 'P/BV*†', 'marketcap_mm',
               'revenue_mm', 'industry']]
sp_df.head()
```

Out[10]:

	company	ticker	price_close	pct_price_change_lastday	pct_price_change_30day	pct_price_change_ytd	pct_price_change_12_month	P/E*†	P/BV*†	marketcap_
0	3D Systems Corporation	DDD	8.21	5.66%	36.61%	(6.17%)	(26.37%)	NM	1.85x	97
1	3M Company	MMM	148.40	(0.13%)	10.92%	(15.88%)	(19.88%)	17.47x	8.41x	8535
2	8x8, Inc.	EGHT	16.11	1.38%	14.58%	(11.97%)	(33.46%)	NM	7.40x	165
3	A. O. Smith Corporation	AOS	41.98	0.79%	17.03%	(11.88%)	(21.74%)	18.75x	4.09x	680
4	AAON, Inc.	AAON	45.41	-	1.11%	(8.10%)	(12.35%)	44.47x	8.15x	236

Remove the Percentage sign and parentheses from pct_price_change

In [11]:
```
pct_price_change_lastday = sp_df['pct_price_change_lastday'].apply(lambda x: x.replace('-', ""))
pct_price_change_lastday = pct_price_change_lastday.apply(lambda x: x.replace('(', '-'))
pct_price_change_lastday = pct_price_change_lastday.apply(lambda x: x.replace(')', ''))
pct_price_change_lastday = pct_price_change_lastday.apply(lambda x: x.replace('%', ''))
pct_price_change_lastday = pd.to_numeric(pct_price_change_lastday)

pct_price_change_30day = sp_df['pct_price_change_30day'].apply(lambda x: x.replace('-', ""))
pct_price_change_30day = pct_price_change_30day.apply(lambda x: x.replace('(', '-'))
pct_price_change_30day = pct_price_change_30day.apply(lambda x: x.replace(')', ''))
pct_price_change_30day = pct_price_change_30day.apply(lambda x: x.replace('%', ''))
pct_price_change_30day = pd.to_numeric(pct_price_change_30day)

pct_price_change_ytd = sp_df['pct_price_change_ytd'].apply(lambda x: x.replace('-', ""))
pct_price_change_ytd = pct_price_change_ytd.apply(lambda x: x.replace('(', '-'))
pct_price_change_ytd = pct_price_change_ytd.apply(lambda x: x.replace(')', ''))
pct_price_change_ytd = pct_price_change_ytd.apply(lambda x: x.replace('%', ''))
pct_price_change_ytd = pd.to_numeric(pct_price_change_ytd)

pct_price_change_12_month = sp_df['pct_price_change_12_month'].apply(lambda x: x.replace('-', ""))
pct_price_change_12_month = pct_price_change_12_month.apply(lambda x: x.replace('(', '-'))
pct_price_change_12_month = pct_price_change_12_month.apply(lambda x: x.replace(')', ''))
pct_price_change_12_month = pct_price_change_12_month.apply(lambda x: x.replace('%', ''))
pct_price_change_12_month = pd.to_numeric(pct_price_change_12_month)
```

In [12]:
```
sp_df['pct_price_change_lastday'] = pct_price_change_lastday
sp_df['pct_price_change_30day'] = pct_price_change_30day
sp_df['pct_price_change_ytd'] = pct_price_change_ytd
sp_df['pct_price_change_12_month'] = pct_price_change_12_month
```

1.3 Determine how many constituents and industries are part of the S&P 1500. Before dropping NAs and after dropping NAs.

Before dropping NAs

In [13]:
```
industries = sp_df['industry'].unique()
len(industries)
```

Out[13]: 148

In [14]:
```
constituents = sp_df['company'].unique()
len(constituents)
```

Out[14]: 1500

After Dropping NAs

In [15]:
```
sp_df = sp_df.dropna()
```

In [16]:
```
industries = sp_df['industry'].unique()
len(industries)
```

Out[16]: 148

In [17]:
```
constituents = sp_df['company'].unique()
len(constituents)
```

Out[17]: 1489

Saved the clean file as SP1500.CSV

In [18]:
```
sp_df.to_csv("../data/SP1500.csv")
```

In [19]:
```
sp_df = pd.read_csv("../data/SP1500.csv",index_col = 0 )
constituents = sp_df['industry'].unique()
```

**The new dataframe for the selected industry: Data Processing and Outsourced Services **

In [20]:
```
data = sp_df[sp_df['industry']=='Data Processing and Outsourced Services']
data.head()
```

Out[20]:

	company	ticker	price_close	pct_price_change_lastday	pct_price_change_30day	pct_price_change_ytd	pct_price_change_12_month	P/E*†	P/BV*†	marketc
56	Alliance Data Systems Corporation	ADS	46.34	-0.09	74.93	-58.70	-70.40	5.40x	2.03x	
138	Automatic Data Processing, Inc.	ADP	144.48	2.89	12.37	-15.26	-9.81	24.05x	11.31x	6
202	Broadridge Financial Solutions, Inc.	BR	113.85	0.70	23.23	-7.84	-3.33	31.46x	11.61x	1
232	Cardtronics plc	CATM	20.72	0.58	24.22	-53.59	-40.66	19.53x	2.43x	
354	CSG Systems International, Inc.	CSGS	47.53	-1.37	15.03	-8.21	2.68	18.75x	3.81x	

1.4 Determine the top 5 worst performing industries and top 5 best performing industries.

Select the industry within more than 10 companies

In [21]:
```
pd.value_counts(sp_df['industry'])[(pd.value_counts(sp_df['industry'])>10)]
```

Out[21]:
```
Regional Banks                          86
Health Care Equipment                   42
Industrial Machinery                    41
Oil and Gas Exploration and Production  32
Semiconductors                          28
Application Software                    28
Property and Casualty Insurance         26
Retail REITs                            26
Packaged Foods and Meats                26
Specialty Chemicals                     25
Restaurants                             25
Biotechnology                           25
Data Processing and Outsourced Services 23
Oil and Gas Equipment and Services      23
Aerospace and Defense                   22
Pharmaceuticals                         22
Specialized REITs                       22
Semiconductor Equipment                 21
Apparel Retail                          21
Electric Utilities                      21
Building Products                       20
Communications Equipment                20
Asset Management and Custody Banks      18
Auto Parts and Equipment                17
Health Care Services                    17
Life Sciences Tools and Services        16
Apparel, Accessories and Luxury Goods   16
Homebuilding                            16
IT Consulting and Other Services        15
Electrical Components and Equipment     14
Electronic Equipment and Instruments    14
Office REITs                            14
Steel                                   14
Multi-Utilities                         14
Construction Machinery and Heavy Trucks 13
Construction and Engineering            13
Trading Companies and Distributors      13
Automotive Retail                       12
Consumer Finance                        12
Thrifts and Mortgage Finance            12
Health Care REITs                       12
Life and Health Insurance               12
Health Care Supplies                    12
Investment Banking and Brokerage        11
Trucking                                11
Health Care Facilities                  11
Specialty Stores                        11
Residential REITs                       11
Systems Software                        11
Name: industry, dtype: int64
```

In [28]:
```
df_picked = sp_df[[k in [*pd.value_counts(sp_df['industry'])[(pd.value_counts(sp_df['industry'])>10)].index] for k in sp_df
```

In [ ]:
```
top5_worst_industries = sp_df_proc.groupby('industry').mean().sort_values('pct_price_change_ytd').head()
```

In [30]:
```
worst = df_picked.groupby('industry').mean().sort_values('pct_price_change_ytd').head()
worst
```

Out[30]:

industry	price_close	pct_price_change_lastday	pct_price_change_30day	pct_price_change_ytd	pct_price_change_12_month	marketcap_mm
Oil and Gas Equipment and Services	9.610000	0.136522	37.949130	-58.038696	-64.260870	2435.592174
Apparel, Accessories and Luxury Goods	28.538750	-1.799375	28.146250	-50.084375	-55.616875	3664.318125
Apparel Retail	17.656667	-2.374762	38.597619	-49.606667	-51.818095	5013.575714
Retail REITs	22.060000	-1.726538	27.491923	-47.069615	-48.809615	3147.117692
Oil and Gas Exploration and Production	16.077813	4.396875	79.428437	-44.879375	-56.579375	5111.217500

In [32]:
```
best = df_picked.groupby('industry').mean().sort_values('pct_price_change_ytd').tail()
best
```

Out[32]:

industry	price_close	pct_price_change_lastday	pct_price_change_30day	pct_price_change_ytd	pct_price_change_12_month	marketcap_mm
Pharmaceuticals	53.501364	0.505000	16.248636	-5.760455	-5.954091	55999.604545
Life Sciences Tools and Services	187.846875	0.383125	19.373125	-2.180000	12.018125	19600.751250
Systems Software	85.375455	1.029091	12.955455	-1.290000	-2.786364	148038.845455
Biotechnology	93.465200	3.387200	20.652000	0.100400	7.686000	24565.139200
Health Care Supplies	101.741667	1.030000	21.951667	5.150000	-4.824167	6107.563333

1.5 Create a new data frame for the worst industry and a new data frame for the best industry both based on % Price Change [YTD]. For each of these new data frames, use the % Price Change [30 Day] to plot the top five ranking for worst and best performance constituents.

Create new data frame for the worst performance industry and the best performace industry

In [25]:
```
df_best = sp_df[sp_df.industry == 'Health Care Supplies']
```

In [26]:
```
df_worst = sp_df[sp_df.industry == 'Oil and Gas Equipment and Services']
```

In [ ]:
```

```

In [33]:
```
sp_worst = sp_df[sp_df['industry'].isin(worst.index.tolist())]
sp_best = sp_df[sp_df['industry'].isin(best.index.tolist())]
```

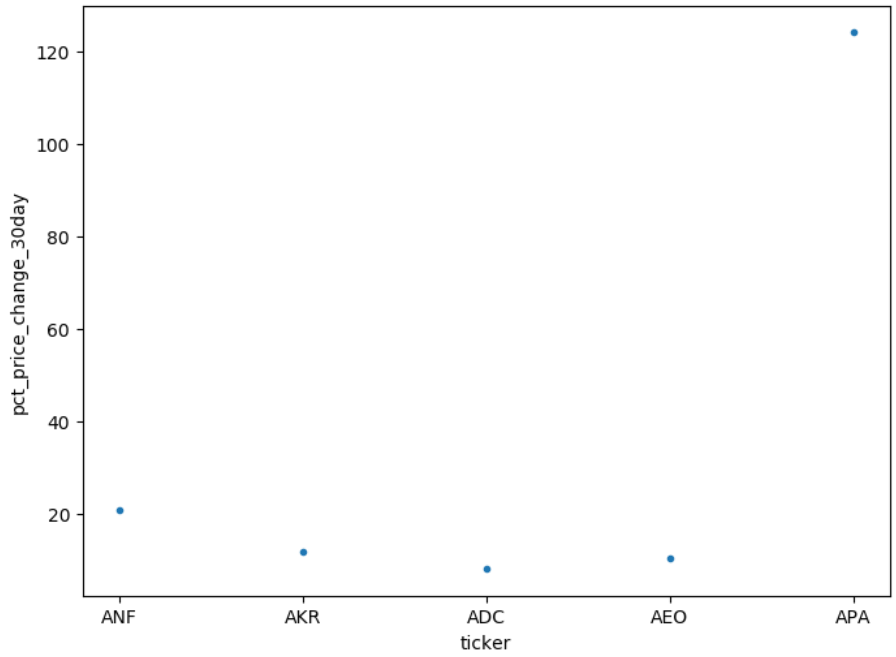
Plot Figures

Top5 for the Worst Industry

In [46]:
```
sp_worst5 = sp_worst.head()
```

In [56]:
```
plt.figure(figsize=(10,8))
plt.figure(figsize=(8,6),dpi=100)
plt.xlabel('ticker')
plt.ylabel('pct_price_change_30day')
plt.scatter(list(sp_worst5['ticker']), list(sp_worst5['pct_price_change_30day']), marker='.')
plt.savefig("../graph/top5_worst_industry.jpg")
```

```
<Figure size 720x576 with 0 Axes>
```



Top5 for the Best Industry

In [54]:
```
sp_best5 = sp_best.head()
```

In [57]:
```
plt.figure(figsize=(10,8))
plt.figure(figsize=(8,6),dpi=100)
plt.xlabel('ticker')
plt.ylabel('pct_price_change_30day')
plt.scatter(list(sp_best5['ticker']), list(sp_best5['pct_price_change_30day']), marker='.')
plt.savefig("../graph/top5_best_industry.jpg")
```

```
<Figure size 720x576 with 0 Axes>
```